

-15-

REMARKS

In response to the Office Action mailed on August 22, 2005, Applicants respectfully request reconsideration. To further the prosecution of this Application, Applicants submit the following remarks discussing patentability of rejected claims 1-35 as set out in the present Office Action.

Applicants respectfully traverse the rejections in the Final Office Action based on grounds that the cited prior art does teach all of the claim limitations and that the Applicants' arguments regarding patentability have been mischaracterized. The following discussion further clarifies the reasons for allowance.

Rejection of Pending Claims 1-30 under 35 U.S.C. §102(e)

The Examiner continues to reject claim 1 under 35 U.S.C. §102(e) as being anticipated by Young (U.S. Patent 6,782,531). The Office Action likens elements in Young to those in claim 1 to reject the claimed invention.

Regarding point (a) made by the Examiner, Applicants respectfully traverse the rejection based on grounds that the cited prior art does not teach all of the claim limitations and that the Examiner mischaracterizes the arguments in the last office action response. For example, the Examiner contends that column 13, lines 60-61 in Young (e.g., "a plug-in will take a long time to execute because it must wait on the results of a database query by another plug-in") reads on the claim limitation of "based on queries to the plurality of plug-in modules, retrieving a dependency list indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of a plurality of plug-in modules" as in claim 1.

First, the claimed technique recites retrieving a dependency list based on queries sent to the plurality of plug-in modules. The passage cited in Young

merely indicates that a plug-in must wait for results of a database query by another plug-in module. Claim 1 recites a technique that operates in an opposite manner as the cited subject matter used to reject the claimed invention. For example, the claimed invention indicates that the queries are sent to the plug-in modules. The cited reference set forth to reject that recites that a “database query” emanates from a plug-in module to a database. The database in Young is not a plug-in module. See FIG. 6B in Young illustrating that the dependency data 654 is stored in execution management framework 652. Note that plug-in modules 662, 664, and 666 reside at a remote location with respect to the execution management framework 652 and that the plug-in modules do not provide dependency information to the execution management framework 652. Thus, based on these grounds, the concepts are not equivalent and the rejection under 35 U.S.C. §102 is improper.

Second, note that the queries to the plug-in modules prompt retrieval of a dependency list, which is defined in the claim as a different kind of information as that recited in Young. For example, according to claim 1, the retrieved dependency list indicates respective plug-in services that are provided by and required by each of the plurality of identified plug-in modules. The “database query by another plug-in module” in Young is not made for purposes of retrieving dependency requirements of or services provided by plug-in modules.

Also, note that corresponding apparatus claim 15 recites that the corresponding steps are executed in a plug-in manager application. There is no indication in the cited passages that a plug-in manager retrieves the dependency and service information based on queries to each of multiple plug-in modules.

This claimed technique enables the plug-in modules themselves to provide dependency information of what other plug-in modules to execute rather than relying on efforts of the execution management framework as in Young to

-17-

track such information for all plug-ins via a configuration file. Also, the plug-in modules provide an indication of what services are supported by respective plug-in modules based on the queries to the plug-in modules. This is also not supported by Young. Thus, Applicants respectfully request the withdrawal of the respective rejection of claim 1 under 102(e).

For the reasons stated above, Applicants submit that claim 1 includes limitations not found in any of the cited references and therefore is patentably distinct and advantageous over the cited prior art. Applicants respectfully request the withdrawal of the rejection of claim 1 under 35 U.S.C. §102(e) or request that the Examiner more particularly point out passages in Young that teach the claimed invention. Accordingly, allowance of claim 1 is respectfully requested.

Because claims 2-14 depend from and further limit claim 1, Applicants submit that claims 2-14 are in allowable condition as well.

Claim 15 includes similar limitations as discussed above for claim 1. Accordingly, Applicants respectfully submit that claim 15 includes similar patentable distinctions over the cited prior art as does claim 1. Applicants respectfully request allowance of claim 15 as well as corresponding dependent claims 16-28.

Applicants submit that claim 3 includes further patentable limitations not disclosed by Young. For example, claim 3 recites "receiving a dependency response from the plug-in module..." The Office Action cites column 13, lines 50-53 to reject this portion of claim 3 of the subject application. The cited language in Young reads as follows:

“The configuration file holds information associated with each node in the graph 600 as needed to execute the plug-in, including information specifying the number of other plug-ins on which each plug-in depends. The stage uses that information in executing the plug-ins in the correct order. At any given time, it might be possible to execute more than one plug-in: if two plug-ins do not depend on each other, the two plug-ins can be executed in any order, or can be executed simultaneously (i.e., in parallel during the same clock cycle) by multiple threads. Therefore, for instance, if a plug-in will take a long time to execute because it must wait on the results of a database query by another plug-in, the pipeline infrastructure can execute other plug-ins in the mean time. At any time, any plug-in that has zero dependencies can be executed.” (emphasis added)

Applicants respectfully submit that this passage provides no literal support or suggestion that any plug-in module as in Young provides “a dependency response from a plug-in module...indicating respective plug-in services provided by, and required by, the plug-in module” as in the claimed invention. That is, the Examiner has not cited language in Young indicating that the “configuration file” resides in a plug-in module. FIG. 6b of Young indicates that the dependency data 654 resides in the execution management framework and not a respective plug-in module. Nor is there any indication in the cited Young reference that any plug-in module itself provides an indication of services supported by the plug-in module.

In a similar vein as discussed for claim 1, the technique in claim 3 enables the plug-in modules themselves to provide dependency information of what other plug-in modules to execute rather than relying on efforts of the execution management framework to track such information for all plug-ins via maintaining a configuration file. Thus, Applicants respectfully request the withdrawal of the respective rejection of claim 3 under 102(e) should be withdrawn.

Applicants submit that claim 5 includes further limitations not disclosed by Young. For example, claim 5 recites “querying a dependency interface ... to obtain the dependency response from the plug-in module.” As recited in claim 3, the dependency response indicates respective plug-in services provided by and required by the respective plug-in module. The Office Action cites column 13, line 61 to reject this claim limitation. This cited passage merely indicates that the pipeline infrastructure can execute another plug-in module if a plug-in module will take a long time to execute because it must wait on the results of a database query by another plug-in module. As discussed above, there is no indication that the database query in Young is generated for the purpose of obtaining dependency information from the plug-in module as in the claimed invention. Nor is the database query executed in Young generated for a purpose of learning which services are supported by a respective plug-in module. The cited passage therefore does not teach or suggest the claimed invention. Thus, Applicants respectfully request the withdrawal of the respective rejection of claim 5 under 102(e) should be withdrawn.

Claim 9 recites “forwarding, via a dependency available interface associated with a respective plug-in module, a list of initiated plug-in services of other plug-in modules that are currently available for use by the respective plug-in module.” To reject this claim, the Examiner cites Young at column 8, lines 31-38 which states:

The stage configuration module passes the plug-in configuration information to an execution management framework 425. The execution management framework 425 uses this information to determine which of the plug-ins nos. 1-8 can be processed in parallel (and during the same clock cycles per clock 420) and which of the plug-ins nos. 1-8 need to be processed in sequence after other plug-ins because they depend on a final or intermediary result from the other plug-ins.

-20-

Applicants traverse the rejection on grounds that the cited passage in Young does not recite the limitations as purported by the Examiner. For example, the claim recites forwarding a list of services to a respective plug-in module indicating which services are available from other plug-in modules. Consequently, the respective plug-in modules according to the claimed invention know which services are available from the other plug-in modules. The execution management framework 425 in Young is not a plug-in module. Moreover, the execution management framework 425 does not communicate such information to any of the plug-in modules. Thus, the rejection under 102 is improper.

Regarding point (C) made by the Examiner, Applicants submit that claim 10 and, more importantly, claim 24 include limitations not disclosed by Young. For example, claim 10 and 24 both recite “forwarding to the respective plug-in module, via a dependency available interface associated with the respective plug-in module, the identity of each initiated plug-in service required by the respective plug-in module.” Claim 24 specifically recites that a plug-in manager application supports operations of “forwarding to the respective plug-in module, via a dependency available interface associated with the respective plug-in module, the identity of each initiated plug-in service required by the respective plug-in module.”

The Office Action cites column 8, lines 31-32 to reject this portion of claim 10 of the subject application. The cited language in Young reads as follows:

“The stage configuration module passes the plug-in configuration information to an execution management framework 425.”

Applicants respectfully submit that this passage provides no literal support associated with forwarding an identity of each initiated plug-in service required by

the plug-in module. FIG. 4 more particularly illustrates that the stage configuration module 416 (see column 8, lines 25-27) passes plug-in configuration information to the execution management framework 425.

The Examiner further cites Young at column 8, lines 30-38 which reads as follows:

The stage configuration module passes the plug-in configuration information to an execution management framework 425. The execution management framework 425 uses this information to determine which of the plug-ins nos. 1-8 can be processed in parallel (and during the same clock cycles per clock 420) and which of the plug-ins nos. 1-8 need to be processed in sequence after other plug-ins because they depend on a final or intermediary result from the other plug-ins.

First, the execution management framework 425 in Young is not a plug-in module. Claim 10 recites forwarding information from a plug-in module manager to a respective plug-in module. Thus, Young does not disclose this aspect of the claimed invention.

Second, contrary to the Examiner's assessment, the execution management framework 425 does not forward an identity of plug-in modules to a respective plug-in module. Instead, as recited in Young, the execution management framework 425 merely determines which of multiple plug-in modules can be processed in parallel and which must be processed in a particular sequence. The execution management framework 425 does not forward any identity information to the plug-in modules. As its name suggests, the execution management framework 425 initiates execution of the plug-in modules nos. 1-8 in a sequence or in parallel by sending the plug-in modules a respective execution command and counter value that is used to determine when to execute the respective plug-in module over time. There is no indication that

-22-

the plug-in modules are aware of the identity of which other plug-in modules are required by the remote server plug-in module.

This claimed technique enables the plug-in modules to manage themselves based on knowing which plug-in module from which they depend. The plug-in modules in Young only understand at what point in time they are to execute and not from which other plug-in modules they depend. Thus, Applicants respectfully request that Examiner allow claim 10 over Young.

Regarding point (d) made by the Examiner, Applicants again submit that claim 13 includes further limitations not found in Young. For example, the claimed invention of claim 13 recites: "wherein the first plug-in module is initiated via the step of initiating operation of plug-in modules after initiation of the second plug-in module, and wherein the second plug-in module includes a wait-state operation causing the second plug-in module to wait to provide the service offered by the second plug-in module until initiation of the first plug-in module."

The passages cited by the Examiner to reject claim 13 include Young at column 13, lines 28-30 which read as follows:

"As noted above, plug-ins can be implemented as modular pieces of code that are executed during run-time for performing a defined task, such as a sub-computation on session data. Usually, the plug-ins need to be executed in a certain, specified order to effectuate the desired, overall computation performed by the stage that contains them. A complexity is introduced in specifying that order because plug-ins can be dependent on other plug-ins. Generally speaking, given two plug-ins M and N, if plug-in M computes the value x (as a final or intermediary result) and plug-in N requires the value x to perform its computation, then plug-in N depends on plug-in M. Plug-ins can be dependent on zero, one, two, or more, other plug-ins. In the above notation, because of the

noted dependency between M and N, the stage infrastructure will wait for plug-in M to be executed before it starts execution of plug-in N. Plug-ins with zero dependencies can be executed immediately or at any other time, without regard to prior execution of other plug-ins.” (Emphasis added)

Based on the cited passage, Applicants submit that the claimed invention operates in a different manner than as mentioned in Young. For example, according to the cited passage, Young indicates that plug-in N depends from plug-in M and that the “stage infrastructure” will wait for plug-in M to be executed before it starts execution of plug-in M. This means that the stage infrastructure executes plug-in M first and “depending” plug-in N later in time. Applicants are unable to follow the logic used to reject the claimed invention.

Applicants again submit that Young discloses a technique opposite to that recited in the claimed invention. First, the claimed invention recites that the “depending” plug-in module (i.e., the second plug-in module as recited in claim 13) is executed first before the non-depending plug-in module (i.e., the first plug-in module as recited in claim 13). This is opposite to the configuration in Young where the “depending” plug-in N is executed after plug-in M.

Second, the claimed invention recites that the second plug-in module includes a wait operation causing the second plug-in module to wait to provide the service offered by the second plug-in module until initiation of the first plug-in module. Young does not recite this technique of waiting because there would be no need to wait; the “depending” plug-in module is executed after the plug-in module on which it depends after data is available. Thus, there would be no need to wait in Young as recited in the claimed invention.

Based on the recited technique in claim 13, there is no longer a need to initiate execution of the plug-in modules in any particular order, even though a

dependency exists. Young makes no mention of initiating execution of plug-ins in the order as described in the claimed invention. Nor does Young discuss a wait type of operation associated with a plug-in to receive services offered by another plug-in. Providing a wait-state operation in Young would serve no useful purpose because it would not be necessary.

For the reasons stated above, Applicants submit that claim 13 includes limitations not found in any of the cited references and therefore is patentably distinct and advantageous over the cited prior art. Applicants respectfully request the withdrawal of the rejection of claim 13 under 35 U.S.C. §102(e) or request that the Examiner more particularly point out passages in Young that teach the claimed invention. Accordingly, allowance of claim 13 is respectfully requested.

For similar reasons as discussed above for dependent claim 13, Applicants submit that claim 27 is also in condition for allowance.

The Examiner provides no particular argument regarding the rejection of claim 29 other than that provided for claim 1. This is an incorrect assessment. Claim 29 includes limitations not found in claim 1.

Applicants would like to point out that claim 29 recites “querying a dependency interface associated with the plug-in module with a dependency query to obtain a dependency response from the plug-in module, the dependency response indicating respective plug-in services provided by the plug-in module.” In short, Young does not disclose querying a plug-in module to learn of plug-in services provided by the plug-in module as discussed above. Applicants respectfully request allowance of claim 29.

Applicants are unable to respond to point (F) made by the Examiner because of the uncertainty to which claim the comment pertains.

Claim 30 includes patentable limitations similar to the limitations of pending claim 13. For applicable reasons as discussed above for claim 13, Applicants submit that claim 30 is patentable over the cited prior art. Claims 31-35 depend from claim 30 and therefore also should be allowable over Young and the other cited references.

Claims 31-35

Applicants submit that dependent claims 31-35 include limitations not found in the cited prior art.

For example, claim 31 recites “querying a set of plug-in modules to identify services provided by the set of plug-in modules.” In short, contrary to the Examiner’s assessment of the cited passage, Young does not disclose that a management entity or any other entity queries any plug-in modules to learn of services performed by the plug-in modules.

For example, claim 32 recites “in response to querying the set of plug-in modules, identifying plug-in modules not identified by the software application as being necessary but which are identified by the set of plug-in modules as being necessary to carry out execution of an operation on behalf of the software.” In short, contrary to the Examiner’s assessment of the cited passage, Young does not disclose that a set of plug-in modules identifies which plug-in modules are necessary to be carried out in order to carry out execution of an operation on behalf of the software application.

For example, claim 32 recites “in response to querying the set of plug-in modules, identifying plug-in modules not identified by the software application as being necessary but which are identified by the set of plug-in modules as being

necessary to carry out execution of an operation on behalf of the software.” In short, contrary to the Examiner’s assessment of the cited passage, Young does not disclose that a set of plug-in modules identifies which plug-in modules are necessary to be carried out in order to carry out execution of an operation on behalf of the software application.

For example, claim 33 recites that “the third plug-in module initiating a wait state operation causing the third plug-in module to wait to provide the service offered by the third plug-in module until instantiation of the fourth plug-in module.” Contrary to the Examiner’s assessment of the cited passage, Young does not disclose that any plug-in modules initiate execution of a wait state operation. Nor does Young recite the other limitations as in claim 33.

For example, claim 34 recites “wherein the processor initiates execution of the first plug-in module before execution of the second plug-in module, the first plug-in module initiating a wait state operation resulting in signaling to the second plug-in module, the signaling indicating that a respective service of the first plug-in module is not yet available to the second plug-in module.” Contrary to the Examiner’s assessment of the cited passage, and for the reasons discussed above, Young does not disclose that any plug-in modules initiate execution of a wait state operation.

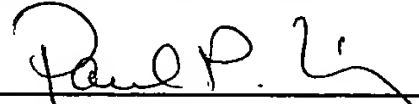
CONCLUSION

In view of the foregoing remarks, Applicants submit that the pending claims as well as newly added claims are in condition for allowance. A Notice to this affect is respectfully requested. If the Examiner believes, after reviewing this Response, that the pending claims are not in condition for allowance, the Examiner is respectfully requested to call the Representative.

-27-

Applicants hereby petition for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-0901.

Respectfully submitted,



Paul P. Kriz, Esq.

Attorney for Applicant(s)

Registration No.: 45,752

CHAPIN & HUANG, L.L.C.

Westborough Office Park

1700 West Park Drive

Westborough, Massachusetts 01581

Telephone: (508) 366-9600

Facsimile: (508) 616-9805

Customer No.: 022468

Attorney Docket No.: EMC01-11(01046)

Dated: October 21, 2005